Argonne NATIONAL LABORATORY

THE UNIVERSITY OF CHICAGO

Programmable Acceleration for Sparse Matrices in a Data-movement Limited World

Arjun Rawal[†], Yuanwei Fang[†], Andrew A. Chien[†‡]

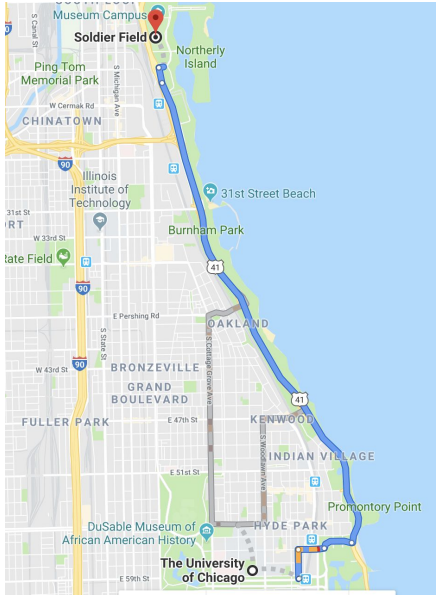
University of Chicago[†]

Argonne National Laboratory[‡]

CERES Center for Unstoppable Computing

Modern Data Processing

- We have more data than ever before
- Data analysis is how we derive insights and value



Data Analysis is Built on Matrix Computation

- Examples include
  - Machine Learning
  - Graph Algorithms
  - Scientific Computing

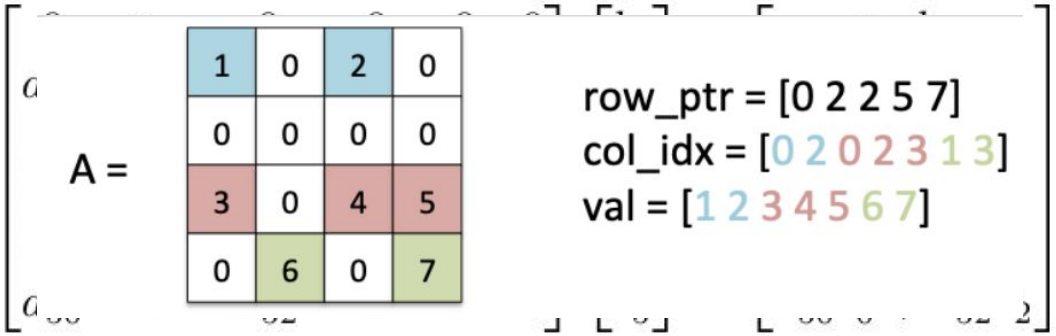
- => All are forms of sparse matrix computation

- Massive opportunity to improve performance on sparse matrices, in particular SpMV (Sparse Matrix Vector Product)

Background

- Custom formats and representations have been a large area of focus
  - COO, CSR, BSR, DIA, ELL ...

- Format optimizations require significant code/data rewrite but performance benefits make it worthwhile



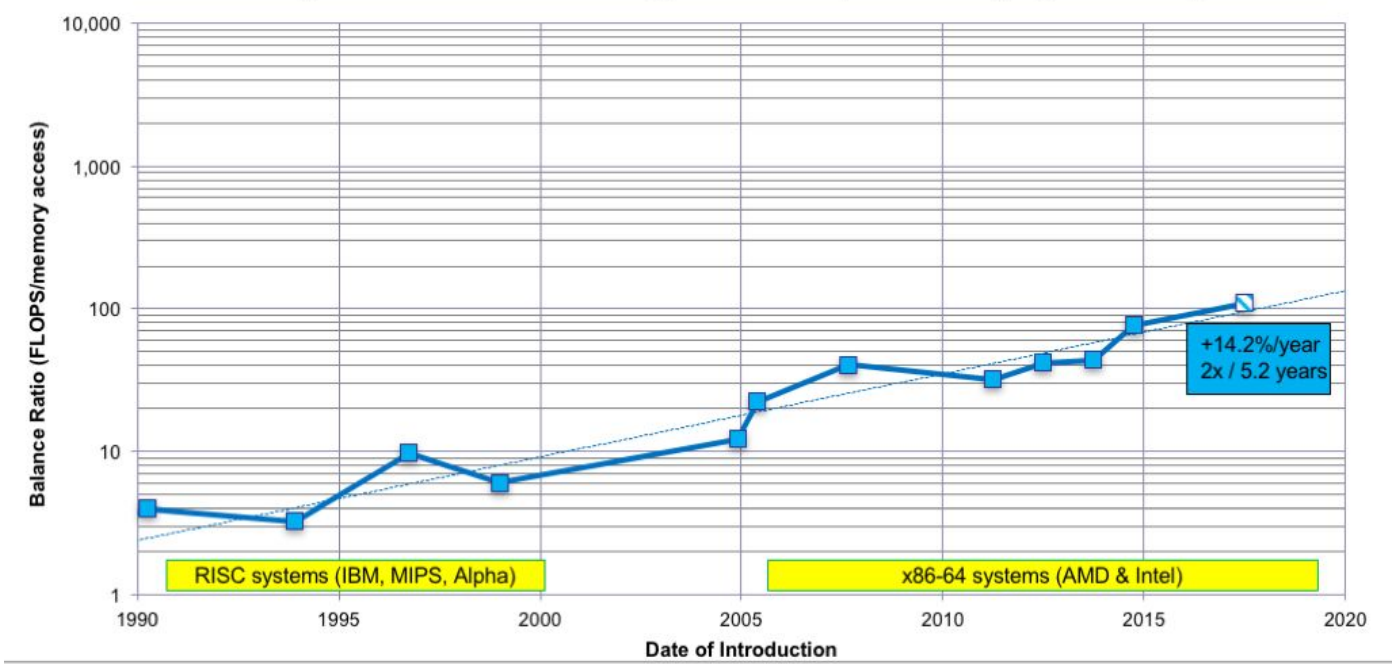
Why is improving SpMV difficult?

- 2 FLOPS/Byte -> low data reuse
- Caching can’t help us here
  - Focus on matrices greater than 1M nonzeros (> 12MB in CSR)
- Memory bandwidth becomes the limit

```
for(int i=0;i<M/B;i++){
 for(int k=i*B;k<(i+1)*B;k++){
   for(int j=row_ptr[i];j<row_ptr[k+1];j++)
        y[i]+=val[j]*x[col_idx[j]];
   }
}
```

CSR SpMV

Compute Performance is Outstripping Memory Bandwidth

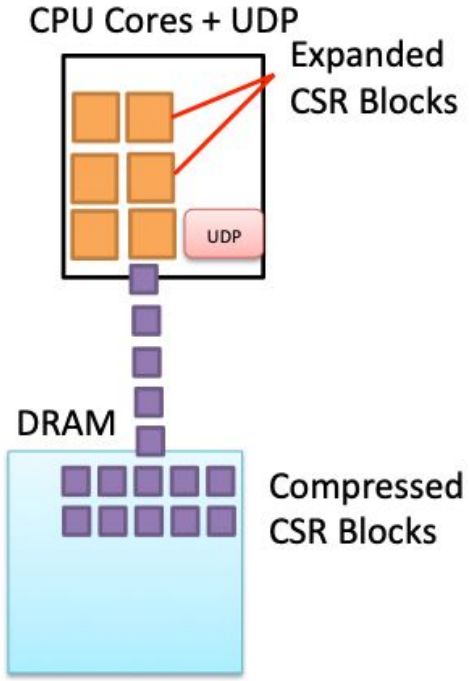


McCalpin's study shows - bottleneck is data movement

Sc16 invited talk: Memory bandwidth and system balance in hpc systems. http://sites.utexas.edu/jdm4372/2016/11/22/

New Approach to Accelerating SpMV

Goal: Use CPU-UDP heterogeneous architecture to overcome memory bandwidth limits through compressed encoding of data

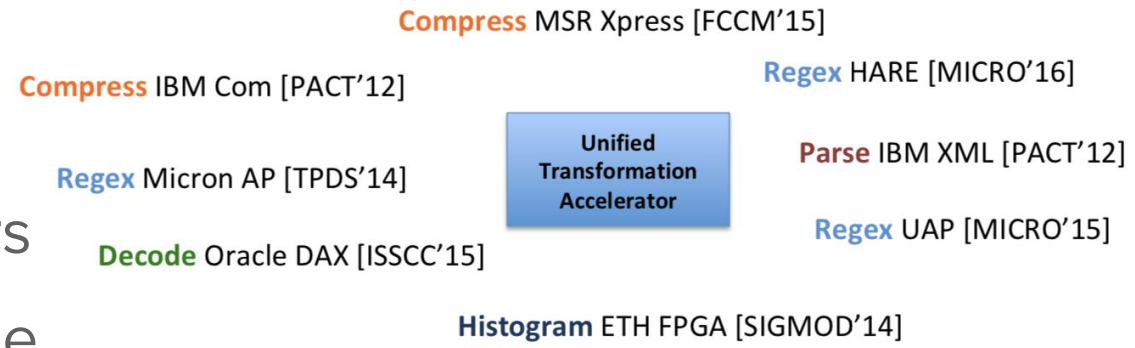


UDP (Unstructured Data Processor)

- We use the UDP , a data recoding accelerator, to help with data transformation (MICRO ‘15, MICRO ‘17)
- UDP is a software programmable accelerator that works with a CPU
- UDP is a low power, high throughput accelerator (tiny)
- Direct memory integration

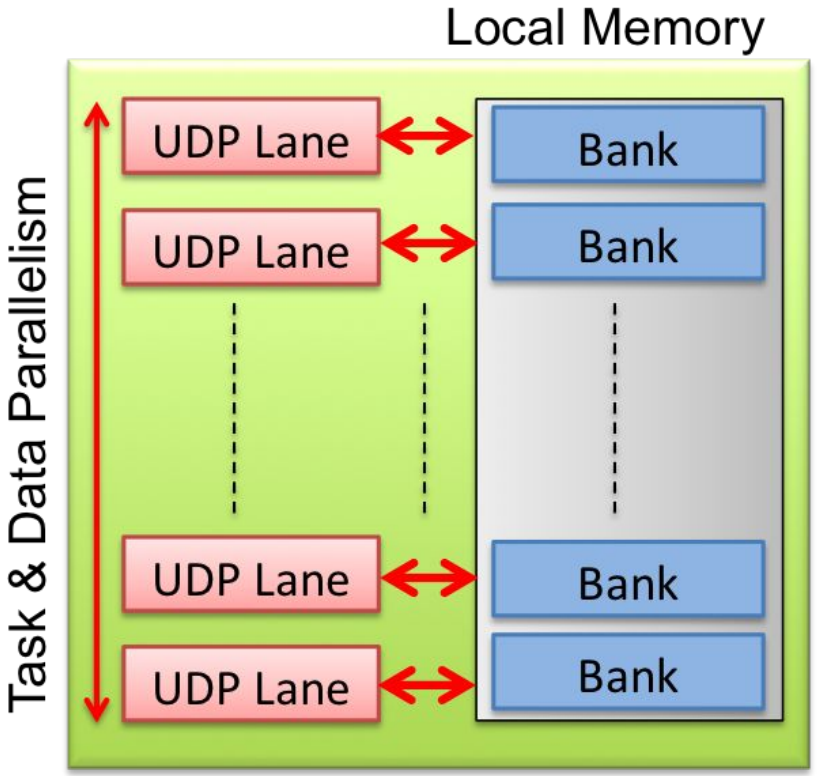
UDP Advantage

- UDP unifies many data processing accelerators
- Software programmable
- Perfect for SpMV because of high throughput, low power, and direct memory integration



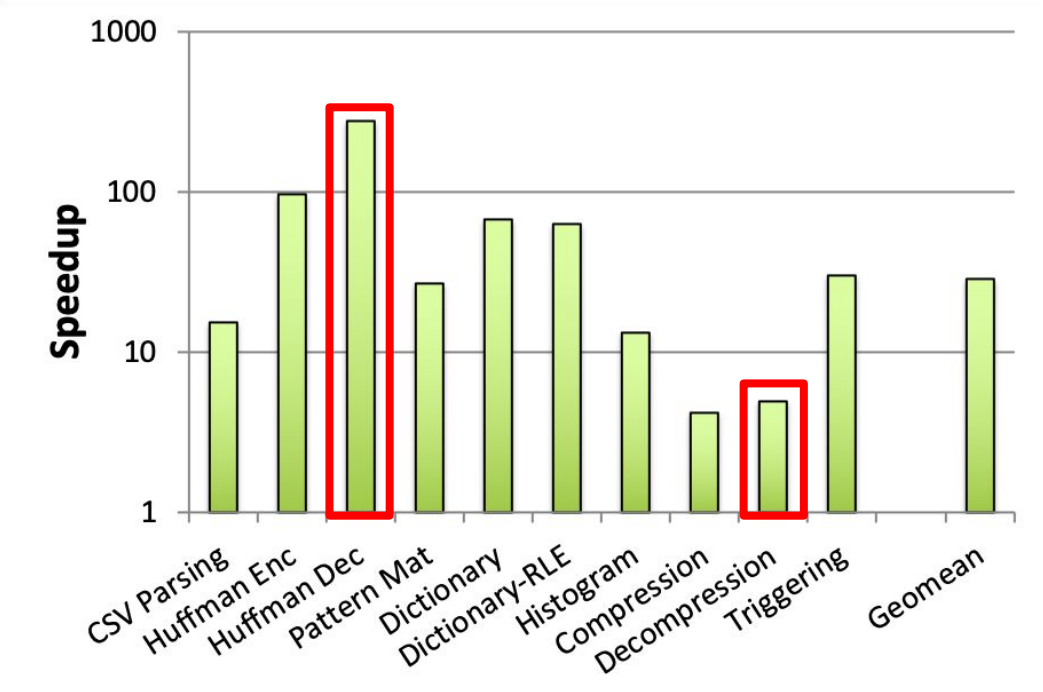
UDP Architecture

- 64 Lane MIMD accelerator
  - Each lane has scratchpad memory and data registers
- 14nm process, 8.7 mm^2 chip area
- 1.6 Ghz and 160 mW
- Used for data transformation workloads that perform poorly on traditional CPU architecture



UDP Performance (1 UDP vs 8 CPU threads)

- >4x speedup on Snappy decompression
- >100x speedup on Huffman Decoding

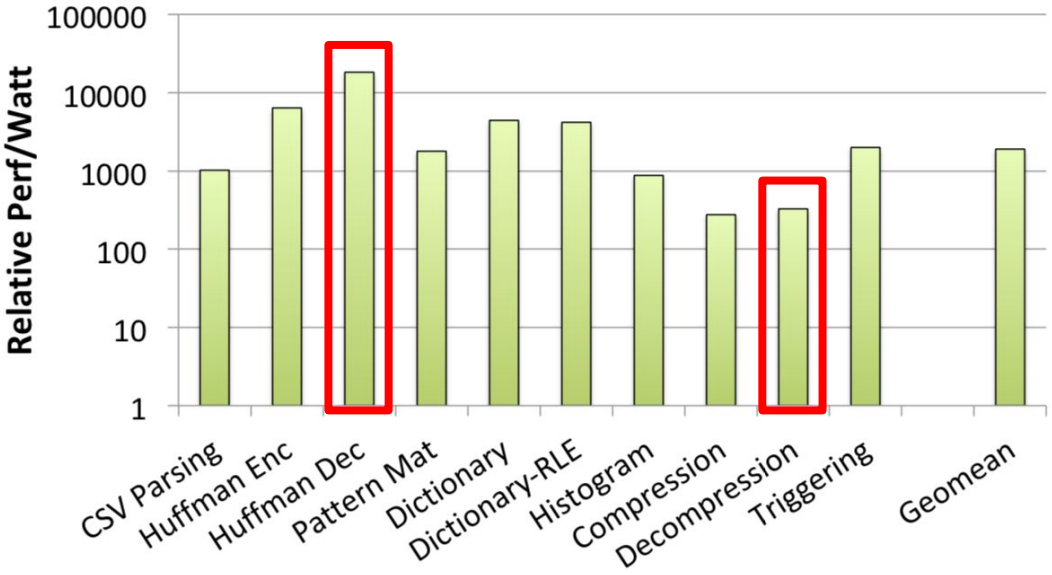


Y. Fang, T. T. Hoang, M. Becchi, and A. A. Chien. Fast support for unstructured data processing: The unified automata processor. In *Proceedings of the 48th International Symposium on Microarchitecture*, MICRO-48, Dec. 2015.

Y. Fang, C. Zou, A. J. Elmore, and A. A. Chien. Udp: a programmable accelerator for extract-transform-load workloads and more. In *Pro- ceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 55–68. ACM, 2017.

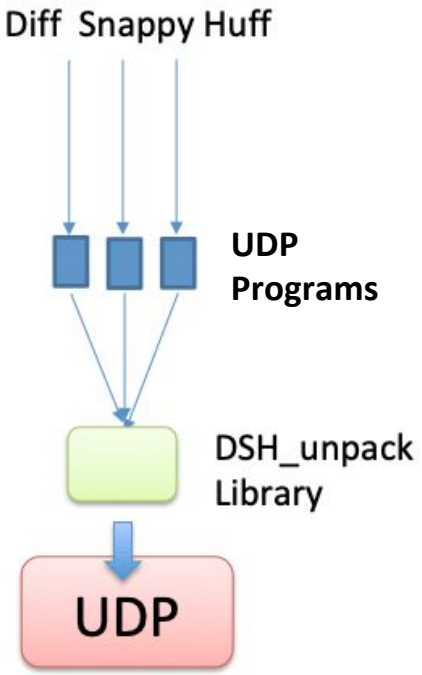
UDP Power Savings (1 UDP vs 8 CPU threads)

- >250x power savings on Snappy on CPU (geomean)
- > 10,000x power savings on Huffman Decoding (geomean)



Data Decompression Workload

- Custom data encode/decode chosen for SpMV on CSR
- Serial Pipeline
  - Huffman Decoding
  - Snappy Decompression
  - Delta Decoding
- Data passes through UDP and is decompressed en route to CPU



Example Integration into CPU Code

```
for(int i=0;i<M/B;i++){
for(int i=0;i<M/B;i++){
 for(int k=i*B;k<(i+1)*B;k++){
   for(int j=row_ptr[i];j<row_ptr[k+1];j++)
        y[i]+=val[j]*x[col_idx[j]];
   }
}
}
```

Evaluation

- Question: Can we decompress matrix data from memory on the fly to improve SpMV performance?

- Dataset: 369 representative matrices from TAMU Sparse Matrix Collection
  - 1M to 8B Nonzeros
  - Sparsity from 9.4E-7 to 19% (Median 0.019%)

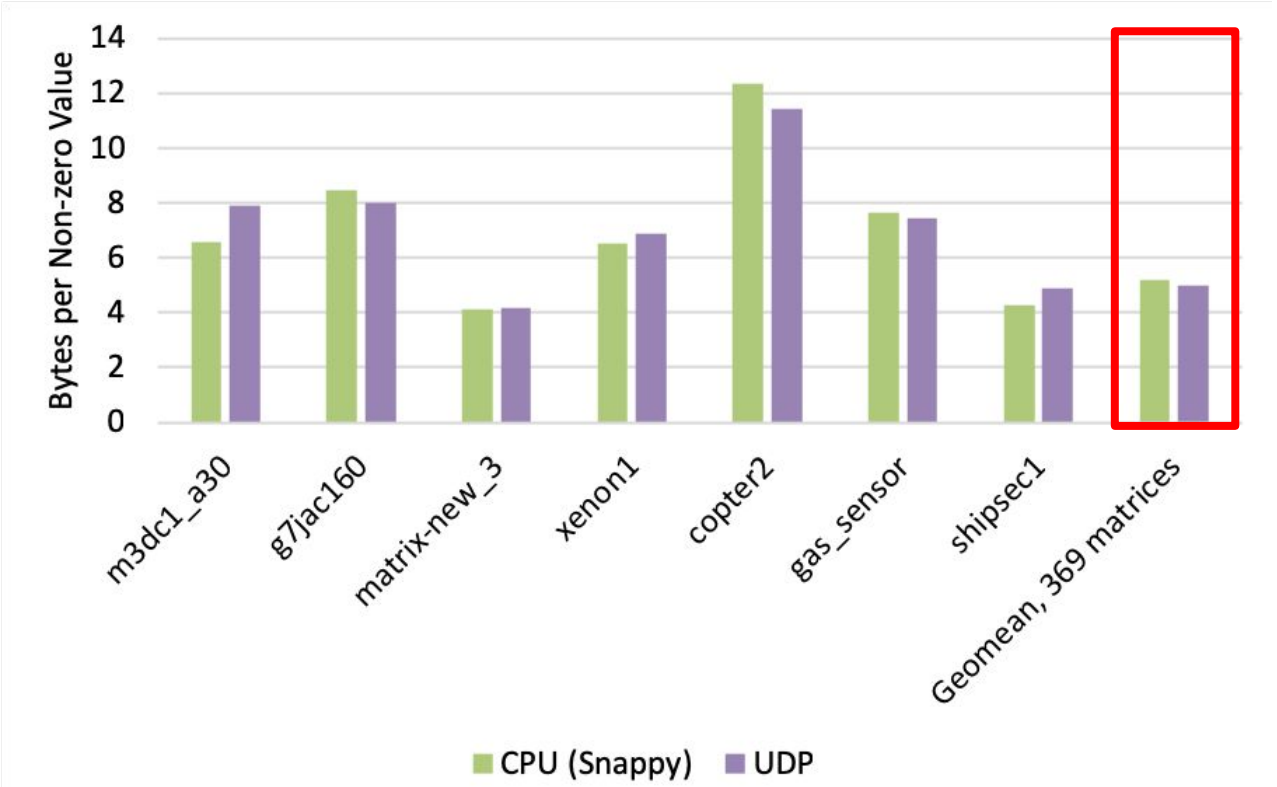
- Data Decompression Workload: 8 KB blocks, parallel, across UDP lanes

Compression Effectiveness on CSR

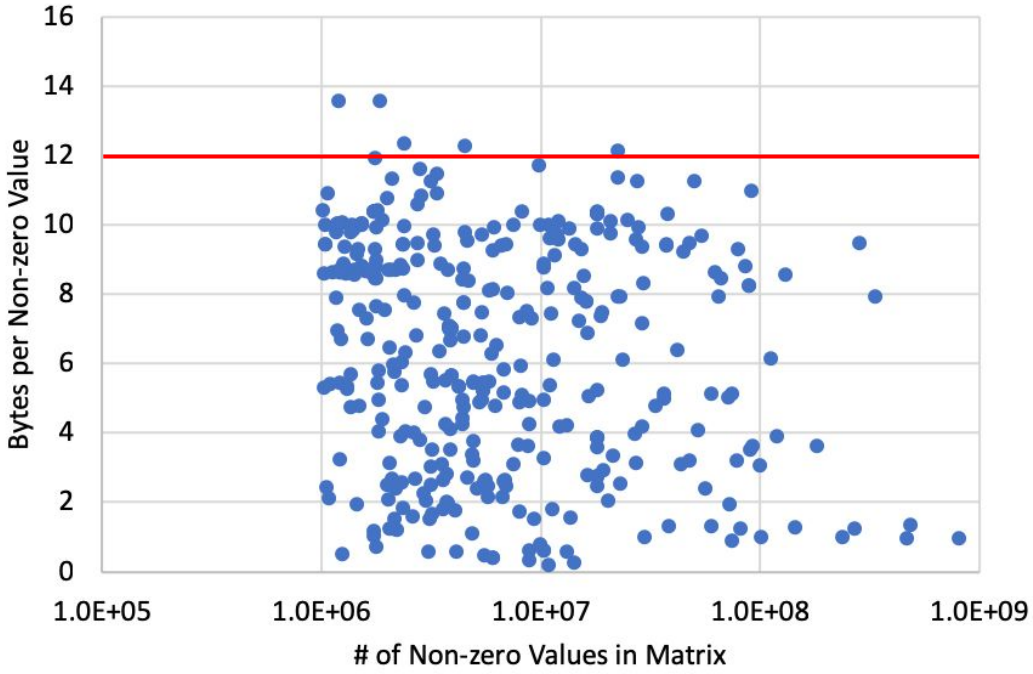
- Highlights 7 challenging matrices, per Department of Energy
- ~7 B/nz for 7
  ~5 B/nz for 369

CSR Nonzero:

Index (4 bytes)	Floating Point (8 bytes)



Compression Effectiveness

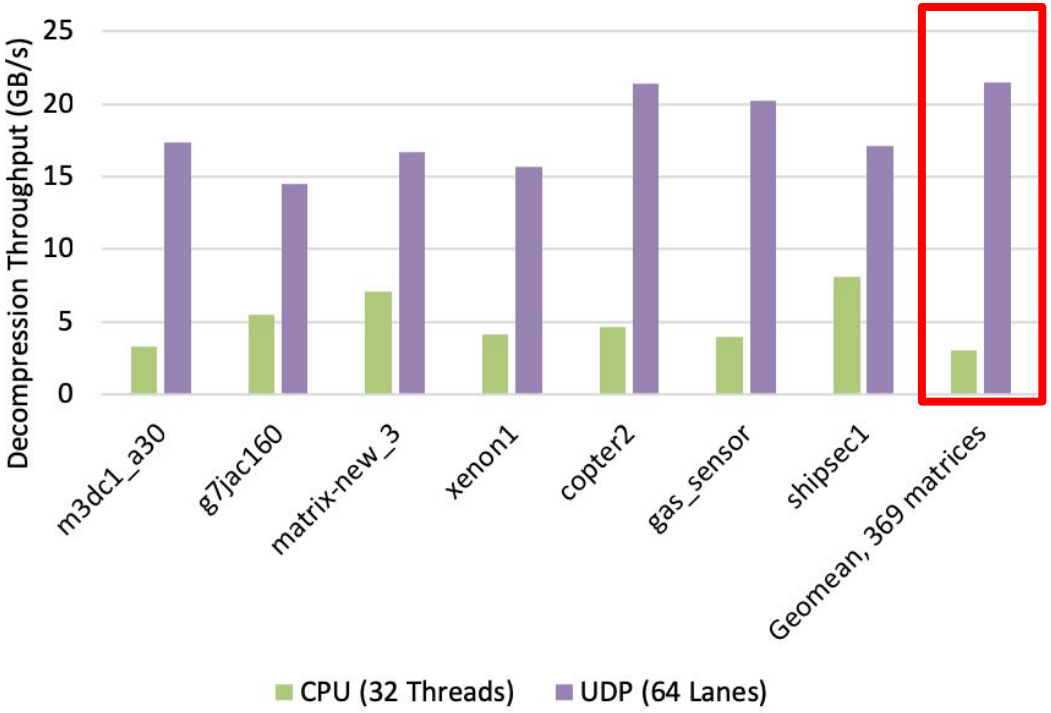


- High variance
- Compression benefits from symmetric and/or banded structure in matrix

- Truly remarkable levels of compression! (< 1 byte per non-zero)

Index (4 bytes)	Floating Point (8 bytes)

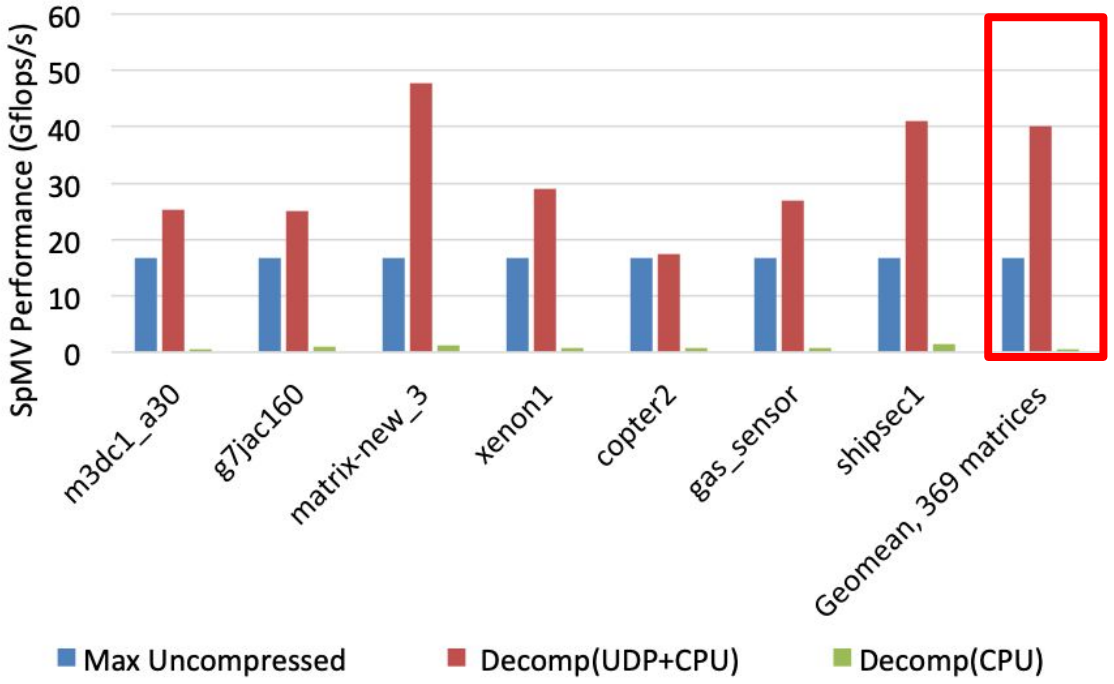
Matrix Decompression Throughput



- UDP does 8KB block in 21.7 μseconds
- Scale across lanes and multiple UDPs
- Provides >6x geomean speedup over 32 thread Xeon CPU

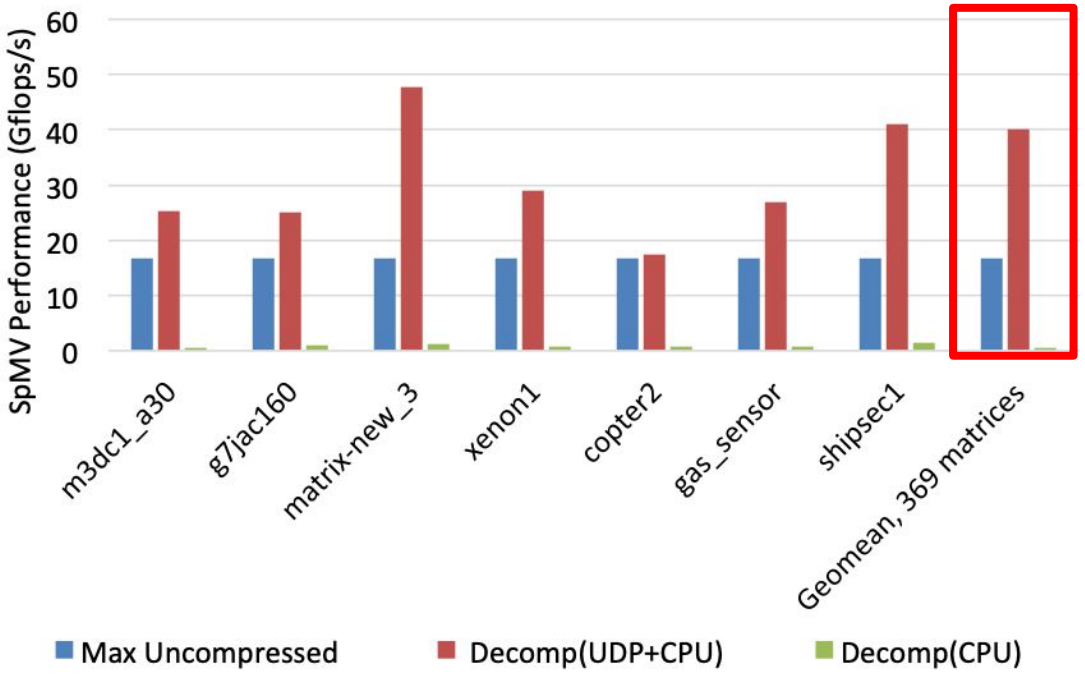
SpMV Performance with DDR4 Memory

- Memory bound computation with 2 FLOPS/nonzero
- Gigaflop increase of 2.4x on dataset
- Faster transfer with DDR4 (100GB/s)



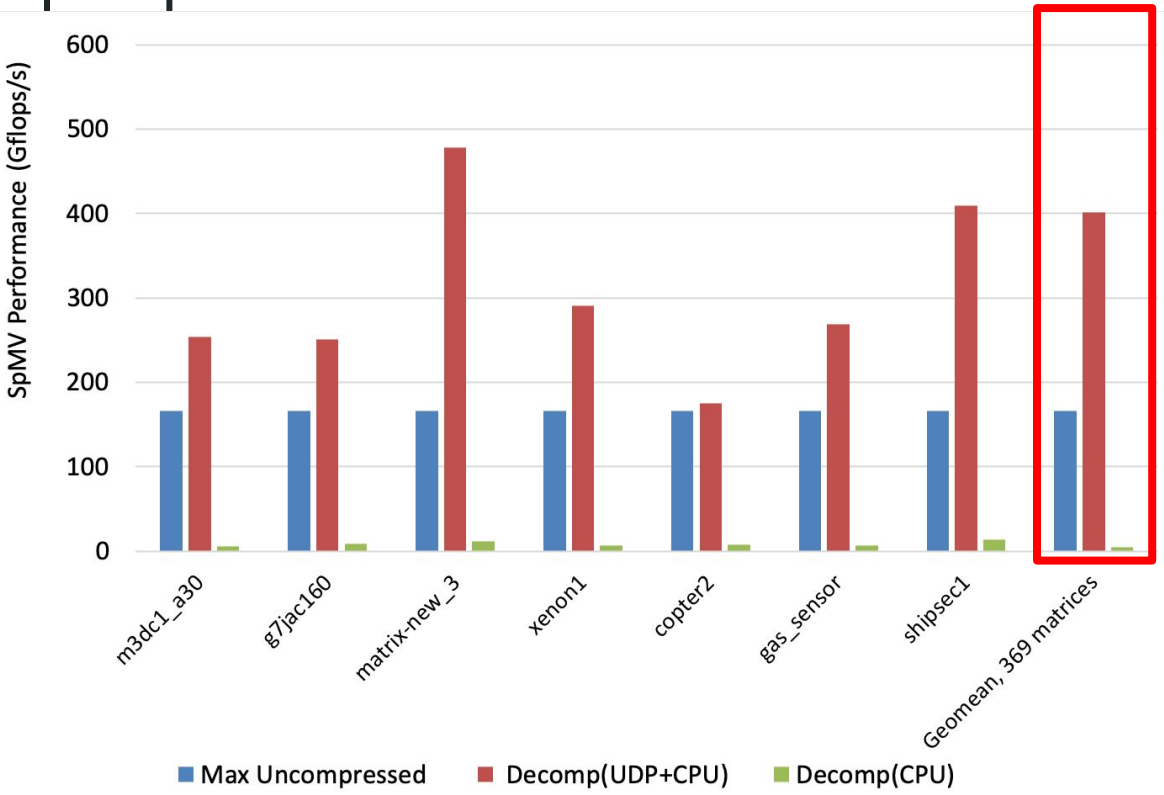
Can we decompress on a CPU?

- No, CPU can’t saturate memory bandwidth
- > 30x slower than CPU-UDP
- Flexible data transformation **not possible** on CPUs



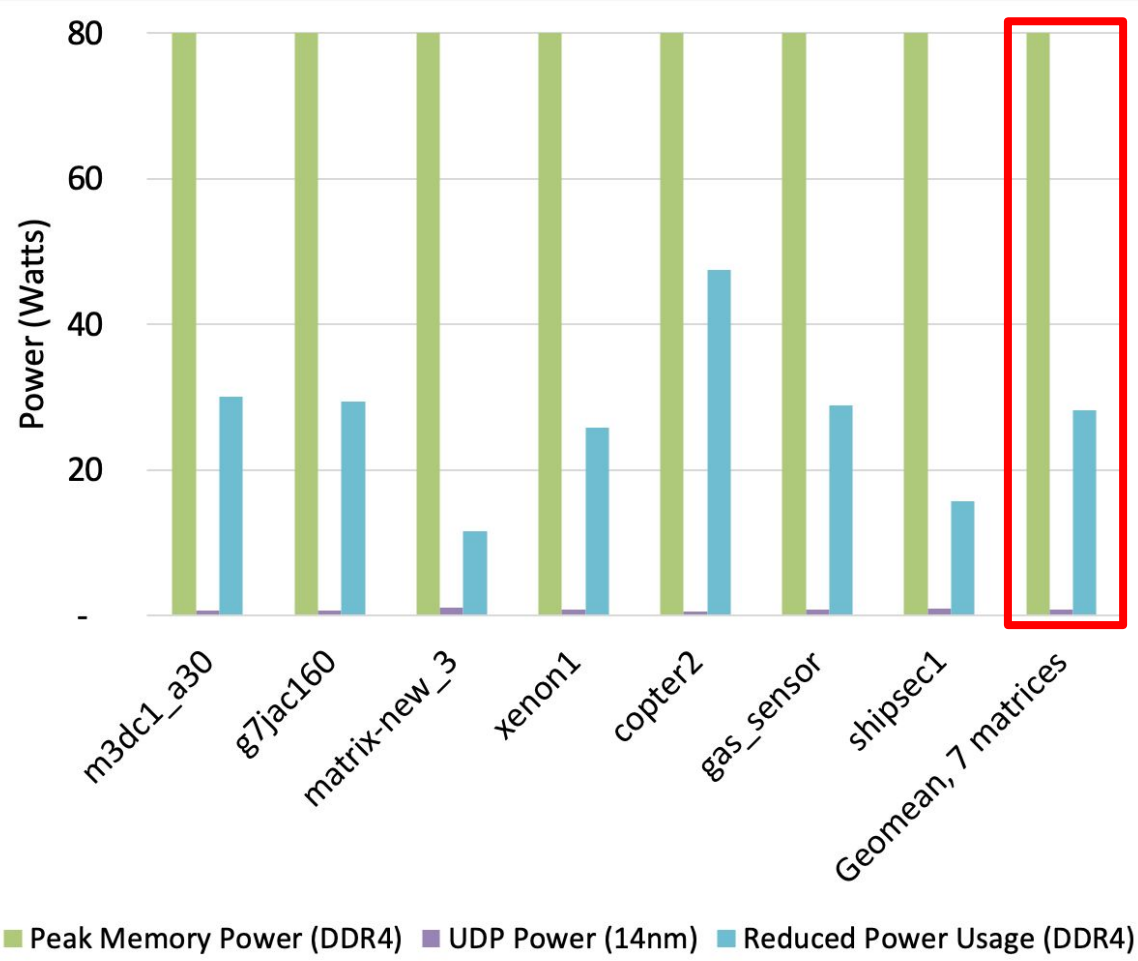
UDP Can Even Keep Up With HBM2

- HBM2 (High Bandwidth Memory) 1TB/s
- UDP able to keep up with HBM2 even at much high Gflops



Power Savings

- Saves 51W of 80W (DDR4, geomean)
- UDP power negligible
- 33W of 64W savings on HBM2 (see paper)



Why use heterogeneous architecture?

- Unproductive to add UDP capabilities to CPU
  - Specialized hardware for specialized applications
  - Software Programmability enables wide variety of use cases

- Huge power savings

- Faster transformation

Related Work

- SpMV Software Optimization (Elafrou, ICPP ‘17)
  - Classifier to choose best optimization based on matrix sample
- SpMV Format Optimization (Kreutzer SIAM ‘14)
  - Unified format for wide SIMD units
- SpMV Hardware Acceleration (Fowers ISCA ‘16)
  - FPGA accelerator to expose parallelism across rows
- Compression Hardware Accelerators (Fowers FCCM ‘15)
  - Pipelined LZ and Huffman accelerator (5.6 GB/s)

Summary

- Choice of matrix encoding can improve performance (moving information, not machine datatypes)
- UDP can recode dynamically at memory speeds (100s of GB/s) and low power usage (< 1W)
- UDP complements CPU with 1800x better energy efficiency on data transformation

Improved SpMV Performance

- 2.4x performance increase, 1600 Gflops/s (HBM2)
- 50% memory power savings at fixed performance (33W - 51W)

Future Work

- SpMV
  - Look into other matrix operations and corresponding storage formats
  - Deep learning training and inference

- UDP
  - Database - faster analytics computations (filtering, parsing, etc)
  - NIC - computation on the wire
  - Storage systems - filtering, real time computations

Acknowledgments

- Prof. Andrew A. Chien, Kevin Fang, and LSSG@UChicago

- This research was partially supported by Exascale Computing Project (ECP) via the Office of Science and the National Nuclear Security Administration

- Additional support from CERES Center for Unstoppable Computing